

BIROn - Birkbeck Institutional Research Online

Yoo, Paul D. (2019) Popularity-based video caching techniques for cache-enabled networks: a survey. IEEE Access 7 , pp. 27699-27719. ISSN 2169-3536.

Downloaded from: <https://eprints.bbk.ac.uk/id/eprint/26766/>

Usage Guidelines:

Please refer to usage guidelines at <https://eprints.bbk.ac.uk/policies.html>
contact lib-eprints@bbk.ac.uk.

or alternatively

Received November 28, 2018, accepted January 8, 2019, date of publication March 4, 2019, date of current version March 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2898734

Popularity-Based Video Caching Techniques for Cache-Enabled Networks: A Survey

HUDA S. GOIAN¹, OMAR Y. AL-JARRAH², SAMI MUHAIDAT³, (Senior Member, IEEE),
YOUSOF AL-HAMMADI¹, PAUL YOO⁴, (Senior Member, IEEE), AND MEHRDAD DIANATI²

¹Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi 127788, United Arab Emirates

²Warwick Manufacturing Group, The University of Warwick, Coventry CV4 7AL, U.K.

³Center on Cyber-Physical Systems, Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi 127 788, United Arab Emirates

⁴Department of Computer Science and Information Systems, Birkbeck College, University of London, London WC1E 7HX, U.K.

Corresponding author: Huda S. Goian (huda.goian@gmail.com)

This work was supported in part by the ICT Fund and in part by the Khalifa University.

ABSTRACT The proliferation of the mobile Internet and connected devices, which offer a variety of services at different levels of performance is a major challenge for the fifth generation of wireless networks and beyond. Innovative solutions are needed to leverage recent advances in machine storage/memory, context awareness, and edge computing. Cache-enabled networks and techniques such as edge caching are envisioned to reduce content delivery times and traffic congestion in wireless networks. Only a few contents are popular, accounting for the majority of viewers, so caching them reduces the latency and download time. However, given the dynamic nature of user behavior, the integration of popularity prediction into caching is of paramount importance to better network utilization and user satisfaction. In this paper, we first present an overview of caching in wireless networks and then provide a detailed comparison of traditional and popularity-based caching. We discuss the attributes of videos and the evaluation criteria of caching policies. We summarize some of the recent work on proactive caching, focusing on prediction strategies. Finally, we provide insight into the potential opportunities and challenges as well as some open research problems enable the realization of efficient deployment of popularity-based caching as part of the next-generation mobile networks.

INDEX TERMS 5G, cache-enabled networking, popularity prediction, proactive caching, videos popularity.

I. INTRODUCTION

The data explosion and persistent user requests have increased pressure on the network backbone. Social media is a growing primary source of data traffic, wherein every minute users around the world post more than 500 million tweets, like ~4.6 billion posts on Facebook, upload ~85 million videos onto YouTube, and watch 5 billion YouTube videos.^{1,2} The end-users demand ultrafast network connections, high data transfer rates, and minimal service delays. The latter influences the retention rate of website users, given that websites may lose more than half of their mobile visitors if the page takes longer than three seconds to load [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Kostas Psannis.

¹<https://socialpilot.co/blog/125-amazing-social-media-statistics-know-2016/>

²<https://fortunelords.com/youtube-statistics/>

This soaring demand requires a paradigm shift towards the development of key enabling techniques for the next generation of wireless networks. In this respect, the fifth generation (5G) network promises to meet the increasing demand for online data streaming by integrating advanced technologies in a cooperative and optimal manner. This includes the development of denser small-cell base stations (SBSs) along with massive multiple-input and multiple-output deployment, beamforming, and broadcasting using millimeter waves. These technologies bring their own complexities and challenges in addition to the inconvenience when scaling up the network, because more base stations and nodes are required in the mobile packet core network [2]. In this regard, edge caching is a favorable method for Internet service providers (ISPs) and content providers because it is less expensive, helps to reduce the overall network traffic and does not require network modifications. Edge caching barter expensive bandwidth resources by the redundant and affordable

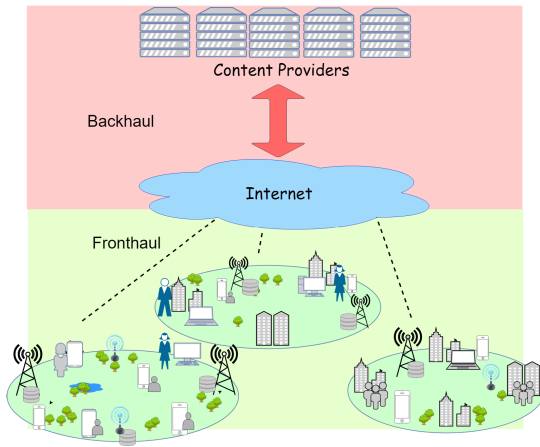


FIGURE 1. Cache-enabled network.

network storage. It transfers content to nodes near the end-users in order to reduce delivery latency and bandwidth usage, lighten the load on the original server, increase the cache hit rate, and improve users quality of experience (QoE) [3].

A. THE VALUE OF POPULARITY PREDICTION FOR CACHING NETWORKS

Cache-enabled networks consist of multiple caching units distributed geographically (Fig. 1). It aims to increase the QoE by removing less popular contents and placing prominent contents in caching units closer to the end-users, such as routers, access points, and base stations. Later, users retrieve popular content immediately without forwarding the request to a further caching unit or the main server. The process of content delivery in case of content available/ not available in the cache is illustrated in Fig. 2. Fortunately, only a few contents are popular and represent the vast majority of network traffic, whereas other contents are rarely requested. For example, 1% of Facebook videos account for 83% of total watch time [4]. Current caching techniques leverage the small number of popular contents and store them, but popularity is

highly dynamic and non-deterministic. A recent trend is to take proactive measures in order to cache future popular contents, leading to so-called proactive caching, which uses popularity prediction algorithms to anticipate user demand and to decide which contents are cached and which are evicted. This enhances overall caching performance and improves replication and delivery strategies. Content popularity also varies spatially. By taking into account the limited number of caching servers in each area, and their restricted storage capability, such caching servers need to serve a substantial number of local users. Thus, each server may store popular content based on the number of requests received per server, but this method causes redundancy in the network as users in the surrounding area manifest similar interests. On the other hand, creating one copy of highly popular content may only shift the traffic from a backhaul link to a fronthaul link. Replication strategies and caching optimization are therefore needed, but network resources such as storage, energy, transmission bandwidth, and computing power are limited, so it is necessary to prioritize contents for caching, to evict less popular content from the caches, and to determine how many replicas are needed and where to store replicas to meet the demands for popular contents [5].

B. IMPORTANCE OF POPULARITY PREDICTION FOR VIDEOS CACHING

The vast majority of today's network traffic produced by streaming video. According to a recent study by Cisco, the mobile traffic generated by videos is expected to reach 75% of the total generated traffic by 2020.³ Videos consume much more cache space than images, documents or emails. Furthermore, the number of videos is increasing because video creation is no longer monopolized by TV channels and companies: most videos now comprise user-generated videos. It is therefore necessary to select popular videos for caching, and popularity prediction can select anticipated on-demand

³<http://tubularinsights.com/2020-mobile-video-traffic/>

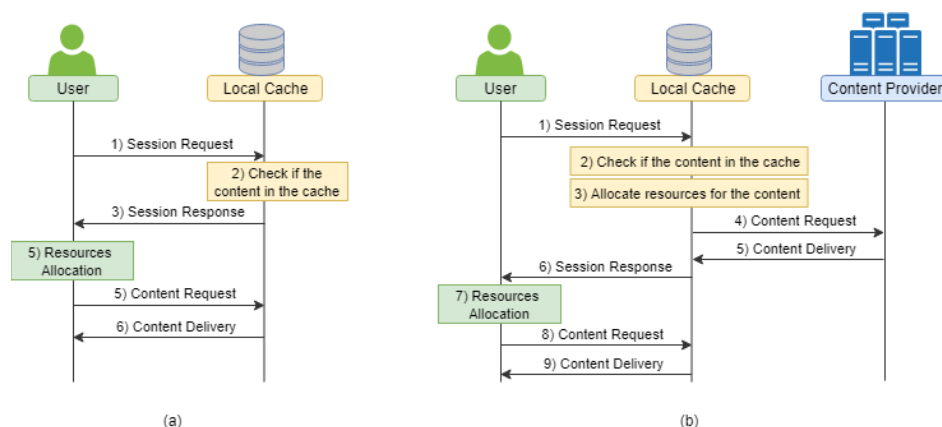


FIGURE 2. Content delivery procedures in small caching-enabled network in case of a) content is not store in the local cache b) content store in the local cache.

videos in order to reduce traffic and better manage caching resources.

Nowadays, users not only care about watching videos but also streaming them with high bit rates and ultra high-resolution [6], [7]. To ensure a great watching experience for viewers, the video must be viewed without stopping or buffering during the video session. Therefore, technologies such as adaptive bitrate streaming (ABR) and dynamic adaptive streaming over HTTP (DASH) are used. These select the best quality based on each user's device and bandwidth, and then dynamically modulate the video quality to maintain a steady playing experience. But, multiple encoded versions need to be created and stored which requires additional processing. Popularity prediction can reduce the additional processing by only focusing on selected, highly-popular videos [4].

C. THE SCOPE OF THIS SURVEY

This paper focuses on the caching based on popularity prediction techniques. In general, video popularity can be measured by various metrics, among which, videos view count is most commonly used. Other metrics include video watch time and rating [8]. Recently, caching has received more attention from the networking perspective. Several reviews of cache-enabled networks have been published, focusing on replica placement techniques [9], caching optimization [10], resource management in the network [11], and energy efficient caching [12], [13]. However, caching schemes based on popularity prediction have not received the same level of attention. A survey in [14] is similar to this study in terms of summarizing popularity prediction algorithms applied to social media content and video. However, the content popularity prediction algorithms discussed are generic and not specifically designed for caching-oriented networks. Another recent survey compressively addresses current prediction and optimization algorithms, including algorithms that predict mobility patterns, resource allocation, and caching [15]. Here, we are concentrating only on anticipating video contents. This paper is summarizing works done in the field of popularity based caching for video contents and address the open research issues in this field.

II. TRADITIONAL VS. POPULARITY-BASED VIDEO CACHING

Current solutions for the optimization of data flow in networks include bandwidth management and compression, but these cannot accommodate the explosive growth in data. One of the key parameters influencing traffic through a network and the corresponding data retrieval response time is the distance between the data centers and the end-user. Therefore, solutions that reduce the required data traveling through the network are preferable. In this context, caching is a promising approach to increase the bandwidth efficiency of wireless networks. Several algorithms that select contents for caching or eviction have been proposed, and these can be classified as traditional or popularity-based caching techniques.

A. TRADITIONAL CACHING SCHEME

Traditional caching schemes assume that current popular content will remain popular. There are several traditional caching policies:

- Least recently used (LRU) is a conventional scheme that has been used in the industry for a long time and can be treated as a baseline to evaluate new caching policies. LRU keeps a record of the last access time for content, and when there is insufficient storage it replaces content featuring the longest idle time with newly requested content. The main drawback of LRU is the use of recentness as a selection metric, because storage may be wasted on unpopular videos just because they were requested recently [16].
- Segmented least recently used by three segments (S3-LRU) divides the cache into three segments organized from the highest popularity in the head of level 3 to the lowest in the tail of level 1. When new content is requested, it is placed in the head of level 1 and the content in the tail of level 1 is removed. If the requested content is available in the cache, it is transferred to the head of the next level, and all the contents below it are shifted downwards [17]. Caching process in S3-LRU is shown in Fig. 3.

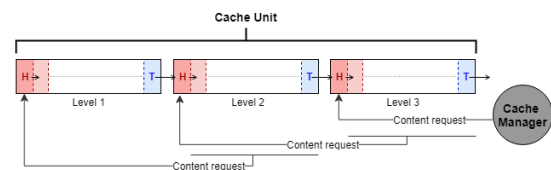


FIGURE 3. Segmented least recently used (S3LRU).

- Least frequently used (LFU) depends on the frequency of requests as an indicator of the popularity of content where the content with the lowest request count is evicted from the cache to make room for new content. LFU suffers from gradual performance degradation because previously-cached unpopular content which received a high number of historic requests remains in the cache, such a phenomenon is known as cache pollution [18].
- Jumping window least frequently used (JW-LFU) adds a predefined window for observing the frequency of requests. The content with the lowest frequency of requests within the window is replaced by content with a higher frequency of requests.
- Sliding window least frequently used (SW-LFU) evicts videos based on the least number of end-user requests. SW-LFU is similar to JW-LFU but it uses a moving window instead of a predefined window.
- The first in first out (FIFO) scheme stores content according to the order of requests, and updates the earliest stored content with new content when not enough storage is available.
- First in first out with least frequently used (FIFO-LFU) considers the request frequency with probability p and

the request order with probability $1 - p$. In this caching policy, FIFO is responsible for caching recent videos that are very sensitive to aging whereas LFU handles videos with stable popularity. The value of p must be selected to maximize the cache hit rate and it varies depending on the category of the video [19].

- Least recently frequently used (LRFU) accounts for recency and access frequency by probability p and $1 - p$ respectively [20].

Traditional techniques are still used in caching networks due to their simplicity and ease of implementation. However, they overlook user behavior and request patterns [20]. Given the limitations in storage and bandwidth, there is a need for better caching schemes that predict highly requested contents and make them available in caching units before peak hours [19].

B. POPULARITY-BASED CACHING SCHEME

Popularity prediction will provide caching schemes with the proactive ability to pick desired contents and make them available in the network nodes. However, this comes with additional computational costs and overheads to guarantee accurate prediction and the execution of correct caching decisions. The desirable algorithm must be quick, capable of handling the large workload, and provide accurate predictions [4]. This section discusses three challenges for popularity prediction algorithms: accurate and reliable predictions, quick predictions, and scalability. We then explore a general framework for popularity-based caching schemes.

1) ACCURATE AND RELIABLE PREDICTIONS

Predicting user behavior is not a trivial task because the future popularity of content is not available as *a priori* information, but it can be measured using parameters such as content quality, historical data, and social interactions. Even so, actual popularity may differ greatly from estimated popularity due to external factors, unexpected situations, the omission of important features in the prediction, or outdated parameters of the prediction model. Inaccurate predictions degrade network performance, and may become even less reliable than traditional caching strategies [21]. Furthermore, popularity distribution patterns differ due to social, spatial and temporal variations. Temporal variations may result in more watching

hours on weekends compared to working days and higher request rates at night. Due to spatial variations, large granularity demand (e.g., a city or town) usually differs from small granularity demand (e.g., a campus or workplace), and different geographical regions may have different popularity distributions. Social variations include changes in social relationships and the joining of new users. Creating models that can observe distribution patterns and updates in terms of spatial, temporal, and social changes is a challenging task [21], [22].

2) QUICK PREDICTION

Some videos experience sudden spikes in popularity soon after uploading, so prediction results must be prepared immediately after uploading to serve the majority of users. However, the results may be inaccurate, due to a trade-off between prediction quality and resolution efficiency. On one hand, fast predictions are needed to maximize resource utilization and user QoE. On the other hand, waiting longer provides more valuable data, allowing more accurate predictions. The prediction algorithm must overcome this challenge and provide a quick forecast with reasonable accuracy in order not to miss out tremendous requests for the most popular videos, just because they were not cached in time.

3) SCALABILITY OF THE ALGORITHM

Scalability grants algorithms the ability to work with high request rates and large-scale data. Content provider workload is exceptionally high, so the algorithm must be scalable to deal with numerous requests while accounting for the limited resources applicable to the algorithm such as the storage for features and learned trends.

4) POPULARITY-BASED FRAMEWORK

A general framework for popularity prediction in caching has been proposed in [22] and [23]. The available social, content and contextual information, such as social relationships, user interest and influence, content demand, server and user geographical location, are the inputs to the system (Fig. 4). The data are then processed to determine the propagation patterns and to gain insight into social, temporal and spatial similarities and differences. Machine learning (ML) is then

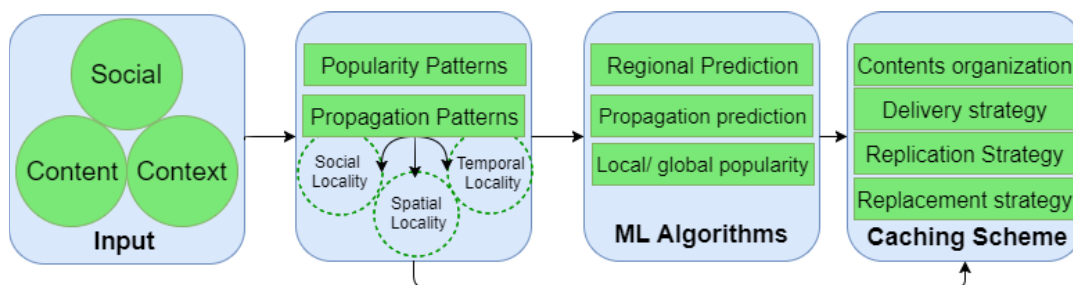


FIGURE 4. Popularity-based caching in machine learning perspective.

applied to predict video popularity and propagation patterns. The performance of the ML model can be improved by using social, temporal and spatial variations, and can be designed to predict where the video will receive more attention in different granularity settings (e.g., local and global). Finally, the ML output is used to plan content replication strategies and to select servers closer to the user.

III. VIDEO POPULARITY PREDICTION ATTRIBUTES

The first block in Fig. 4 shows informative data inputs for prediction algorithms, and these are known as features. Features help to determine the causes of video spreading, and can be broadly classified into four main groups: static, temporal, cross-domain and social features (see Fig. 5).

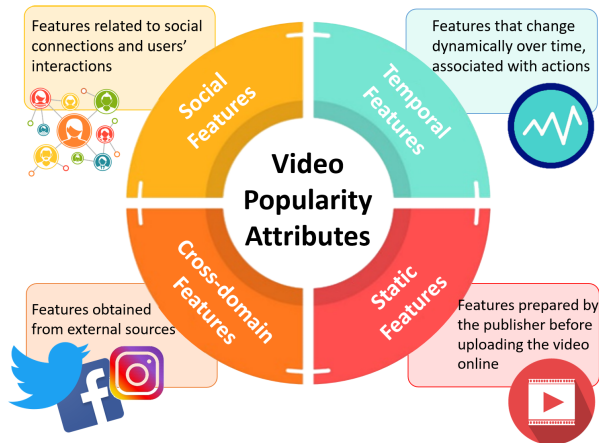


FIGURE 5. Video popularity attributes.

A. STATIC (INTRINSIC) FEATURES

Static features refer to features that are usually fixed and prepared before publishing the video online. It is divided into:

1) VIDEO CHARACTERISTICS

including video duration, the number of frames and dimensions, video and audio quality, mood and genre for music videos, the publication date, and audio features. Audio features refer to features obtained from the background effects and voices in the video. Some attributes can be used to compute higher-level features. For example, the extraction of low-level features such as pitch and tempo from the most representative segment in the music clip can be used to estimate the missing mood and genre entries (high-level features) in the dataset before making the prediction [24].

2) VISUAL FEATURES

Videos comprise a sequence of images, and image processing can therefore be applied to each frame. Visual features include:

- Color, such as finding the dominant color in every frame.
- Face, such as the number of faces detected per frame, the number of frames including faces, and the size of the face regions with respect to the frame size.

- Text, such as the area containing text and the number of frames including text.
- Thumbnail, that is the image that appears in search engine or video sharing results. A study by Netflix revealed that the selection of suitable thumbnail images can increase views by up to 30% [25]. Trzcinski and Rokita [26] extracted 19 attributes from thumbnail images, including the quality, illumination, contrast, overexposure, resolution, and entropy of the thumbnail, etc.

There are also other visual attributes such as scene dynamics and rigidity.

3) TEXTUAL FEATURES

Many features can also be derived from the title of a video, the keywords, category and the provided description. Appropriate preparation of textual information contributes to an easier finding of the video through search engines and therefore increases the view counts.

- Title: This can be used to extract features such as the number of words, punctuation marks, letters, nouns, verbs, adjectives, adverbs, prepositions, numbers, English and non-English characters, Google hits and title words in a list of popular title words, in addition to the sentiment of the title.
- Keywords and descriptions, such as words count, length, and sentiment.
- Category including the popularity of the category and its expected age [26], [27].

B. TEMPORAL FEATURES

Temporal (dynamic) features refer to information that changes over time, associated with actions such as loading the video, posting comments and rating the video. Temporal features provide more useful information than static features. The number of views is the most important temporal feature [28]. Also, the larger the time series data the more accurate the prediction. Temporal features are categorized as follows:

1) CHANNEL FEATURES

From the metadata, features such as the number of subscribers, shares, views, raters, and comments can be obtained for the current time, since the video was uploaded, or for a previous time window.

2) VIDEO FEATURES

Previous channel features can be collected for videos, such as the number of subscribers driven by the video. Video features also include video watch time, number or probability of playbacks, video age, and the anticipated lifetime of the video. The lifetime of a video is the number of days between publication and the day the view count falls below an adequate view count level [29]. Video age is the time since the video is published. Videos are age-sensitive because recent videos tend to attract more viewers than old videos, and the

golden period of a video is typically early in its lifetime. Furthermore, the level of age sensitivity varies among video categories. For example, news and sports videos are much more age sensitive than music videos [19].

3) REQUESTS

The stored requests of clients can be used to find the number of requests and the request rate for a certain video [30].

C. CROSS-DOMAIN FEATURES

Cross-domain features are variables obtained from external sources that can be used to forecast future content demand. They are one of the main causes of sudden spikes in popularity. Such features use data describing the user's reputation on social media to predict the popularity of videos uploaded by the same user onto a video streaming site. For example, features from Twitter can be used as an input for prediction models to determine the popularity of YouTube videos [25].

It is important to note that videos can go through various phases, including spreading, experiencing a sudden burst of popularity, and finally losing user attention. The initial spreading is mainly caused by viral behavior due to subscribers watching the content. This is followed by steady growth due to a migration effect, where other viewers find the video by searching or because of advertisements or placing on the recommended list. Later, external factors can refresh the popularity at any time even after video extinction, such factors include real-life actions and propagation of the content via social media [25].

D. SOCIAL FEATURES

Online social networking is a fascinating and powerful tool. Opinions, ideas, and thoughts can cross to several geographical areas and communities with a single click through social media connections. Twitter and Facebook retain a key influence in disseminating information, thus exploiting social media predictors can help to predict the future of content within the social network (single domain) and outside it (cross-domain). Social network actions, such as posting, sharing, rating and commenting, cause the content to be forwarded to other users followers. Social features include, but are not limited to:

1) FOLLOW

Social interactions can provide insight into the future popularity of content. Usually, social connections are modeled by social graphs, where users are nodes and relationships among them are edges. The direction of edges is used to differentiate followers from followees. Social graphs not only provide follower and followee quantities, but also shows users encouragement level.

2) DEGREE

- In-degree: refers to the number of edges directed toward the node and it represents the number of followers.

- Out-degree: is the number of edges spreading out from the node and it shows the number of followees.
- Total-degree: is the sum of in-degree and out-degree and it shows the user's prestige.

3) CENTRALITY

indicates the importance of a node or user and it includes:

- Betweenness centrality: This quantitatively measures the importance of a user in linking other users in the social media.
- Closeness centrality: The farness of a node sums up the distances between the node and all extant nodes in the social graph. The inverse of farness is the closeness centrality [31].

Many features can be used to predict video popularity, but some are correlated or do not offer descriptive information for the algorithm. Because multiple correlated features provide similar information, including them all actually slows the algorithm down where only one representative feature might be significant. Furthermore, the inclusion of irrelevant features reduces the accuracy of the model by introducing bias. Therefore, the careful selection of informative features is important because it reduces the computational time and cost, improves the prediction accuracy, achieves better data interpretation, and removes the abundant and redundant attributes. This can be achieved by using feature selection algorithms.

Feature selection methods can be divided into three categories: filter, wrapper, and embedded. Filter methods use the training data and rank the features by their relevance. Feature relevance indicates the usefulness of the feature, and it can be measured in several ways including the use of correlation, mutual information, and F-statistic. However, some attributes are more useful when associated with others via ML algorithms. Filter methods can only analyze individual features, whereas wrapper methods consolidate the designed model and pick a subset of features that maximizes an objective function. This involves the use of searching algorithms, where features are added and removed until the maximum objective function is achieved. Such algorithms can take a long time, whereas embedded methods limit the computational time required when testing the subsets by making intelligent selections.

An important consideration when choosing the feature selection algorithm is stability. In a stable selection algorithm, features remain descriptive to the output even when new training samples are introduced into the dataset [32], [33].

IV. PERFORMANCE EVALUATION METRICS

A prediction algorithm that learns user behavior and detects dynamic changes in popularity is integrated into the caching scheme. Therefore, the effectiveness of popularity-based caching depends on the performance of the algorithm and its impact on the caching network. Evaluation metrics can be categorized as algorithm-centric or cache-based metrics, which are summarized in Fig. 6. In this section, we discuss the two evaluation metrics in detail.

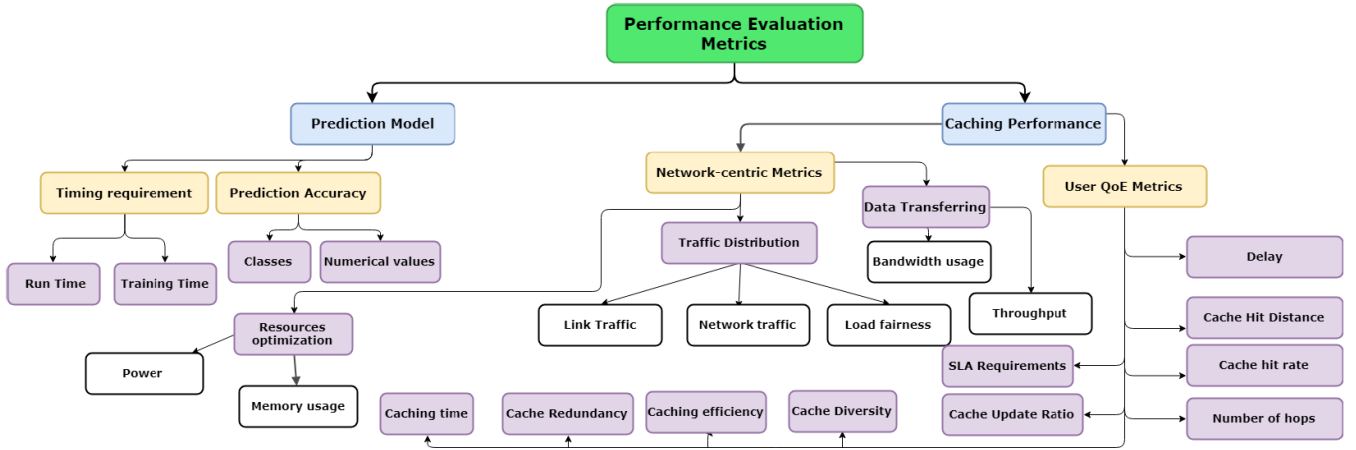


FIGURE 6. Performance evaluation metrics.

A. ALGORITHM-CENTRIC METRICS

1) PREDICTION ACCURACY

different evaluation metrics can be used, depending on the type of problem (classification/regression).

- Classification algorithms: usually assign content to one of two categories (popular and unpopular), or three categories (high, moderate, and low popularity). Accuracy is the most well-known metric to evaluate the performance of a classifier because it shows the ability of the classifier to predict the correct class of an unknown data point. The accuracy of classification model can be calculated as follows:

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

where, in the case of two popularity-based classifiers, true positive (TP) and true negative (TN) are the numbers of correctly classified popular and unpopular videos, respectively, whereas false negative (FN) and false positive (FP) are the numbers of incorrectly classified popular and unpopular videos, respectively.

Although accuracy is often used as an evaluation metric for classification problems, accuracy might not be a suitable metric to evaluate the performance of a classifier for an imbalanced dataset because it assigns equal costs to misclassifications in different classes [34]. Confusion matrix and receiver operating characteristics are potential alternatives of classification accuracy. Confusion matrix shows the TP, TN, FP and FN values for each class in a table format, whereas the ROC method provides a graphical representation of the binary classifier performance by plotting TP versus FP rates for every possible threshold. ROC can be quantified by finding the area under the ROC curve, or AUC. Other metrics based on information retrieval can also be used for imbalanced data, such as recall, precision and F-score,

which combines recall and precision.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F_score = \frac{2Recall \times Precision}{Recall + Precision} \quad (4)$$

- Regression algorithms: model the relationship between the independent variables (inputs) and the dependent variables (outputs). For numerical predictors, accuracy reveals how closely the predictor can estimate the exact numerical values. One of several metrics for assessing regression models is the mean absolute error (MAE). It explains the inaccuracy of the model by finding the average of the sum of absolute errors, where the error is the difference between the estimated and the actual value, and it can be computed using the equation below:

$$MAE = \frac{1}{N} \sum_{t=1}^n |e_t| \quad (5)$$

Another metric is the mean squared error (MSE), which represents the average value of square errors.

$$MSE = \frac{1}{N} \sum_{t=1}^n e_t^2 \quad (6)$$

By taking the square root of the MSE, the resulting scale is equivalent to the output variable scale. This metric is known as root mean squared error (RMSE), which compared to the MAE and other regression metrics tends to assign higher weights to large errors [35].

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^n e_t^2} \quad (7)$$

Low MAE, MSE and RMSE values are desirable. On the other hand, R-squared metric, which indicates the goodness of a regression model, should be high because with

higher values the instances get closer to the fit.

$$R^2 = \frac{ESS}{TSS} \quad (8)$$

where ESS and TSS refer to the explained sum of squares and the total sum of squares respectively [36].

2) TIME REQUIREMENTS

Prediction algorithms are applied during low-traffic load periods, after a predefined time period, or when the content is requested. Due to the number of requests, the algorithm must be suitable for real-time applications by making fast decisions. The testing time is the time consumed by the model to predict the new incoming contents. In real-life scenarios, the parameters of the prediction model must be updated because the prediction accuracy might degrade and the model would not keep up with dynamic user behavior. In this sense, the training time, which denotes the time to train and build a new prediction model using new data, is used to measure the time required to update a prediction model [37].

B. CACHE-CENTRIC METRICS

Caching aims to enhance the user QoE and to improve network performance and utilization.

1) NETWORK-BASED METRICS

These metrics are used to evaluate the performance of the network by measuring the efficiency of traffic distribution, resource management, and data transmission.

- **Bandwidth saving and throughput:** Bandwidth in computing refers to the maximum amount of data that passes between the sending and receiving ends, also known as the maximum throughput. If content desired by the user is cached near to the receiver end, data transfer through backhaul links is reduced. This increases bandwidth saving and reduces network throughput [38].
- **Load fairness:** measures how fairly is the distribution of resources and data among nodes in the network. It also helps to identify overloaded nodes. Fair resource allocation not only equalizes the transmission speed and bandwidth gain among nodes in the network, but also improves resource utilization and reduces redundancy and starvation [39].
- **Link traffic:** identifies links that suffer from extreme traffic and can measure the amount of traffic on each link.
- **Network traffic:** represents the total traffic in the network and can be computed by summing all links traffic values in the network.
- **Energy cost:** Reducing power consumption is one of the major challenges in caching networks. Transmission and caching energy are the two main contributors to energy consumption. The transmission energy is dissipated at the core, edge, and access networks. Edge caching reduces the transmission energy because content is stored closer to the end-users. However, a single

caching unit usually serves limited users, which requires the storage of multiple replicas of popular content in different caching units, thus increasing the caching energy. The caching energy is consumed when reading or writing content, and depends on the caching hardware technology. Although the cost of storage has declined significantly in recent years, read/write operations cannot occur when the memory cells are idle. The greater the quantity of active contents, the more energy dissipation [40].

- **Storage usage:** is consumed by the contents, replications, and the multiple video resolutions that might be saved to increase the data transmission speed [7].

2) USER QoE-BASED METRICS

is divided into the following:

- **Cache hit rate:** is the number of requests delivered by the server divided by all requests. If the content is found in the cache, it produces a hit, but otherwise a miss. This is an important metric to evaluate proactive caching and high values of cache hit rate are desirable.
- **Cache hit distance:** Instead of measuring the availability of the content on the server, the cache hit distance finds the distance from the content to the user divided by all requests. Reducing this parameter improves the quality of service [41].
- **Caching efficiency:** refers to the ratio between the number of requests served by the cache and the total contents in the cache.
- **Cache diversity:** shows the diversity in the network caches by counting the number of distinct stored contents [41].
- **Cache redundancy:** Whereas cache diversity ignores replication, this metric counts the number of copied content within the network. The availability of content in multiple nodes allows requests to be directed to other nodes if the closest node is loaded. Increasing this metric significantly reduces the cache diversity and affects the cache hit rate [42].
- **Caching time:** refers to the time that a content remains stored in the cache [43].
- **Cache update ratio (cache eviction rate):** defines as the number of contents substituted by more popular contents over the total stored contents. Lower cache eviction rate mitigates the power dissipation but increasing this metric may save storage for popular contents by quickly removing the unpopular ones [8].
- **Latency or delay:** refers to the amount of time spent from requesting the video until the frame is displayed. Delay reduction is important especially for live video streaming to provide a real-time display for the viewer. Unlike the non-streaming contents, videos need to be buffered continually until the end of the streaming session [38].
- **Hop-count:** is measured by counting the number of nodes between the server containing the content and

the user. Reducing it can lower the latency and reduce the traffic [43].

- Service-level agreement (SLA) requirements: Meeting SLA requirements is one of the constraints for service/content providers because they need to provide a certain level of quality to the end user [44].

V. POPULARITY-BASED CACHING ALGORITHMS

This section discusses popularity-based algorithms adopted in the field of network caching. They are divided, according to the features used, into two classes: single domain and cross-domain algorithms. Throughout this section, popularity refers to the view count unless popularity definition is stated. The methods are summarized in Table 1.

A. SINGLE DOMAIN

Single domain denotes methods utilize only features from the video sharing site, and it is divided into three categories:

1) POPULARITY EVOLUTION TRENDS

The high correlation between the past and future popularity of a video indicates that historical information can reflect the upcoming trend [45]–[47]. Therefore, popularity growth graphs charting video popularity for consecutive days can be used to anticipate future popularity. In this category, all the methods use time series data (e.g., view or request counts) with different lengths of time series and various future prediction points.

Xiaoqiang *et al.* [48] exploited a first-order gray model which comes from system control. The gray model predicts future popularity values using past popularity sequences collected over different periods, and a new sequence generated by accumulated generating operation (AGO). A first-order differential equation that links both sequences is then built and solved to determine content popularity. Subsequently, reasonable caching locations are selected for each content by combining betweenness and predicted content popularity. Betweenness identifies nodes with multiple distribution paths or high network traffic, and reflects the node influence in the network. To effectively utilize the available cache size, the most popular content is stored in highly important nodes, so that the popular content is closer to a larger number of clients.

Instead of predicting the popularity of the whole video, the popularity prediction algorithm can be applied to each chunk in the video. Actually, chunk-based prediction offers more accurate content popularity than the object-based prediction [49] at the cost of higher computation expense. Zhang *et al.* [50] proposed a chunk-based cache replacement method that leverages the relationship among chunks in the same video stream. They found that requests of the neighbor chunks can be used to forecast the popularity of upcoming chunks and it can be calculated by linear weighted combinations of requests count of previous chunks. Nevertheless, newly requested chunks (e.g., the first chunk of the video) has no records about surrounding ones. Therefore, the prior

requests counts were used in the prediction. All the nodes were equipped with popularity prediction model, and the resulted popularity was compared with the popularity of stored chunks.

However, the previous two methods simulate the network with generated requests that follow Zipf's distribution. The main problem with Zipf's distribution is its inaccuracy in the context of small populations of edge caching units. Convenient performance evaluation therefore requires real traces, where the content request arrival times are non-stationary, extremely heterogeneous and not based on common assumptions [51]. Moreover, effective selection of time series size was not taken into account in [48] but it is actually essential considering abundant historical records in the network. The more extensive the time series, the more knowledge can be derived about the prominence of content requests. However, the computations are enormous considering the input vector and content quantity, and the computational power and memory required to predict popularity for all contents [52].

To reduce computations overhead and effectively utilized the time series information, Nakayama *et al.* [52] proposed an auto-regression model which uses sufficient durations of time series data for accurate prediction while reducing the network cost. To reduce the network cost, data was divided among three states: prediction target (PT), candidate target (CT), and selection target (ST). Prediction algorithm was only performed to contents in the PT state. In the beginning, PT was initialized with data stored in the cache. Content evicted from PT was sent to ST. Then, the LRU returned contents that become popular to CT, whose purpose was to fill the empty space in the PT. Both the CT and PT monitored the number of requests for contents included in their target and kept the corresponding time series data, unlike ST where previous view counts were not recorded in order to save memory. However, the evaluation of [52] did not consider the computational time and cost to illustrate the gain from the created states. Li *et al.* [53] also consider reducing the computational cost. They built a neural network model to analyze TV program request patterns and predict the next-day popularity. Popularity in this work is defined as the request count for particular program divided by the mean request count for all shows. Having analyzed patterns of daily views before selecting the input vector, they found that more view counts occurred at weekends compared to weekdays. Then, they limited the input vector to the popularity of two consecutive days and the total popularity during past seven days.

Network providers have strict SLA contracts where meeting them may result in wasting caching resources. Silvestre *et al.* [54] proposed Hermes, a scheme capable of adeptly determining the number of replicas for popular videos and meeting SLA restrictions. Two support vector machine (SVM) models were designed. The first SVM classified popular videos, then the second SVM processed popular content and classified the videos as increasing, decreasing,

TABLE 1. Overview of the popularity-based caching methods discussed in this survey.

Work	Prediction Method	Features or inputs	Data source	Performance evaluation Metric	Remarks
[48]	Gray Model	Historical popularity for N period	-	<ul style="list-style-type: none"> Average hop distance Average cache hit ratio Content delay Server load strength Network stress 	<ul style="list-style-type: none"> Includes node selection for CCN Simulation in CcnSim via request follow Zipf's distribution
[50]	Linear regression	<ul style="list-style-type: none"> Assist-predict : number of request of surrounding chunks Self-predict: previous number of request of the chunk 	-	<ul style="list-style-type: none"> Average delay Average cache hit ratio Average server load Average cache hit ratio 	<ul style="list-style-type: none"> Studies the relation among chunks in ICN Simulation in NDNSim via request follow Zipf's distribution Improvement of evaluation metrics compare to CCP, FIFO, LRU, and LFU
[52]	Auto regression	5 historical values of the access frequency of the content	-	<ul style="list-style-type: none"> NMSE Cache hit rate Hop counts Eviction rate of unused contents 	<ul style="list-style-type: none"> Includes reduce prediction target in ICN Chunk based simulation in Ccn-Sim Higher cache hit rate by around 1.6 times compare to LRU
[53]	Neural network	<ul style="list-style-type: none"> Popularity(t-1) Popularity(t-2) sum of popularity(t-1) to popularity(t-7) 	Historical IPTV logs from ZTE Corporation	<ul style="list-style-type: none"> Accuracy Time hit ratio 	<ul style="list-style-type: none"> Requires offline training The proposed method can achieve accuracy above 90% and Time hit ratio above 95%
[54]	Two-stage support vector regression	Popularity growth curves	Popularity growth curves obtained from real YouTube logs	<ul style="list-style-type: none"> Prediction accuracy Meet SLA requirements in term of min bit rate, and cache and network storage Bit rate 	<ul style="list-style-type: none"> Considers finding the number of replicas of contents in demand Designed for Hybrid CDNS and simulated on PeerSim Deals with firm SLA contracts Improvement in bitrate can reach about 90%
[55]	DES and Basic experts	<ul style="list-style-type: none"> Today solicitation Yesterday solicitation The calculated solicitation at time t 	YouTube growth data	<ul style="list-style-type: none"> Cumulated loss per phase and the overall 	<ul style="list-style-type: none"> Not tested in caching scheme
[56]	k best experts forecaster	View counts	YouTube growth data	<ul style="list-style-type: none"> Cumulated loss Maximum instantaneous loss Best rank 	<ul style="list-style-type: none"> Online learning predictor
[57]	k best experts forecaster (ARMA models)	View counts	Real traces extracted from the YouTube CDN	<ul style="list-style-type: none"> Hit Ratio Update Ratio Best rank and prediction error 	<ul style="list-style-type: none"> Improvement in hit ratio and update ratio compared to LFU
[58]	Linear/ Power law/ Exponential/ Gaussian models	12 historical request counts recorded every hour	Video on Demand service of a leading European telecoms operator	<ul style="list-style-type: none"> Absolute prediction error Execution time Cache hit rate 	<ul style="list-style-type: none"> Computational cost was not consider Increases cache hit rate by 20%
[59]	Transfer learning	Historical access data	-	<ul style="list-style-type: none"> Cache hit rate Average transmission cost 	<ul style="list-style-type: none"> Achieves better performance than randomized replacement, LRU, non-cooperative learning based caching strategy Content placement is based on greedy algorithm

TABLE 1. (continued.) Overview of the popularity-based caching methods discussed in this survey.

Work	Prediction Method	Features or inputs	Data source	Performance evaluation Metric	Remarks
[16]	Clustering	Number of requests in the previous 5 hours, 30 hours, 5 days, and 30 days	movie.douban.com	<ul style="list-style-type: none"> Running speed Cache hit rate 	<ul style="list-style-type: none"> Online approach and requires no training stage Dose not need training stage Better cache hit rate than FIFO, LFU and LRU
[20]	Linear regression	<ul style="list-style-type: none"> Watch time Number of shares Features were selected based their correlation with view counts	YouTube data obtained via YOUStatAnalyzer ⁴	<ul style="list-style-type: none"> Average Hit Ratio Accuracy metric: MSE/R-squared/Cp Execution time Memory usage 	<ul style="list-style-type: none"> Better performance than LRU, JW-LFU, SW-LFU and FIFO-LFU
[18]	Random forest	<ul style="list-style-type: none"> Request timestamps, user, and program metadata User and program meta-data 	Dataset from one of the Portuguese IPTV operator	<ul style="list-style-type: none"> Backhand data transfer Run time Hit ratio 	<ul style="list-style-type: none"> Outperforms LRU, LFU, and FIFO
[61]	Neural network	<ul style="list-style-type: none"> Metadata with information about multimedia type, style, theme, popularity before releasing, the target group,the cost, etc. PCA was used as feature selection method	Historical download logs collected in Beijing University of Posts and Telecoms	<ul style="list-style-type: none"> ANT ratio Time to manage the cache Relative prediction error 	<ul style="list-style-type: none"> Achieves lower ANT ratio compared to LRU and LFU
[62]	Revealed preferences	Metadata such as :title, description, author, thumbnail, subscribers, viewer, comment, and rate	YouTube Data obtained using YOUStatAnalyzer	<ul style="list-style-type: none"> Average local costs 	<ul style="list-style-type: none"> Adapted learning and selection of caching contents
[17]	Extreme learning machine	Large set of metadata which was reduced using wrapper features selection	YouTube traces obtained using YOUStatAnalyzer	<ul style="list-style-type: none"> Downloading delay Hit rate TP and TN Training time Newtork operating cost 	<ul style="list-style-type: none"> Simulation were done using NS-3 simulator Online learning
[63]	Echo state networks	Content requesting time, week, gender, job, age, and type of users equipment	Real traces from Youku and University of Beijing ⁵	<ul style="list-style-type: none"> Prediction accuracy Design complexity Computational overheads 	<ul style="list-style-type: none"> Caching scheme for cloud radio access networks Needs information about the users'and their requests Better sum effective capacity compared to random caching with clustering and without clustering
[4]	Neural network	<ul style="list-style-type: none"> Approximation of historical features such as comments, likes, shares, and saves counts over four kernels with various time windows. metadata such as the video uploader like and friends count 	Accesses logs of Facebook obtained from Scribe ⁶ .	<ul style="list-style-type: none"> CPU overhead Encoding overhead Future watch time rate of the selected videos 	<ul style="list-style-type: none"> Online learning scheme Scalable and requires only 4 machines to handle Facebook workload Provides predictions every ten minutes Lower the encoding CPU by 3 time

⁴<http://www.congas-project.eu/youstatanalyzer-database>⁵Data are available at: <http://index.youku.com/>⁶Facebook Scribe: <https://github.com/facebook/scribe/wiki>

TABLE 1. (continued.) Overview of the popularity-based caching methods discussed in this survey.

Work	Prediction Method	Features or inputs	Data source	Performance evaluation Metric	Remarks
[68]	SI model + inter-arrival (staleness) approach	Requests rate and Social features	YouTube traces	<ul style="list-style-type: none"> • Hit rate 	<ul style="list-style-type: none"> • Achieves 14% improvement over LRFU
[51]	Discrete-time Markov chain with SIR model	Social features	Facebook/ Twitter data	<ul style="list-style-type: none"> • Run time • MRSE 	<ul style="list-style-type: none"> • Requires neither training phases nor prior knowledge
[71]	Two of three-layer feed-forward neural networks	Video microblogs: Number of root/ re-share/ influenced users and the geographic distribution of video microblogs in the previous M days where Max m=7	From Youku and Tencent Weibo	<ul style="list-style-type: none"> • Bandwidth reservation • RRT 	<ul style="list-style-type: none"> • Considers where to cache (geo distribution) and replicates • Used features are depending on the numbers not features about their quality and influences • Long input feature set which results in memory consumption
[72]	LDA+FE	YouTube: The average category life time/ View count in the last 6 hours. Mainstream media: popular keywords	Data from YouTube and Mainstream media	<ul style="list-style-type: none"> • Local cache hit ratio • Normalized average delay 	<ul style="list-style-type: none"> • NS-3 simulator was used for simulation. • No timing analysis of how long the popular topics selection takes • Offline approach • Shows improvements over LFU, LRU and random caching
[73]	Fast threshold spread model (FTSM)	Facebook: Average number of posts, shares, and comments by user + the number of likes for each of the user's posts + the relationship among users	Data were retrieved using Facebook API and YouTube API ⁷	<ul style="list-style-type: none"> • Accuracy evaluated using ground truth statistics of the respective YouTube video 	<ul style="list-style-type: none"> • Requires neither training phases nor prior knowledge

⁷Data are available at: <http://index.youku.com/>

or remaining constant in popularity. The popular content was cached via the first classifier with the required number of replications determined by the second classifier.

However, the methodologies proposed in [53] and [54] were prepared in offline stage without considering updating parameters. This may later result in performance degradation as contents will be selected by outdated model. Also, prediction models (especially ML-based models) rely heavily on a large training dataset that requires extensive memory for storage and may be affected by noise, resulting in underfitting or overfitting.

Many studies have used online prediction to change the model or update its parameters, in order to cope with popularity distribution continuous variation. The work in [55] assessed three experts for popularity prediction: single exponential smoothing (SES), double exponential smoothing (DES) and the basic expert. SES and DES exponentially smooth past observations to compute the future prediction based on a smoothing parameter. In contrast, the basic expert

does not depend on any tuning parameters, but instead adds the difference between the current and previous solicitations to the current one. The authors argued that videos in the same category experience a similar evolution pattern and can be divided into several phases. To detect phase changes automatically, the basic expert has been utilized as it quickly discovers phase changes. Then, the next solicitation value was found via DES because it resulted in the lowest cumulative loss per phase among the three experts. Hassine *et al.* [8] integrated popularity prediction into a caching delivery network platform to prove the ability of experts to improve caching performance. First, the best expert was selected depending on the sum of rewards for each video. Reward is defined by the number of times an expert minimizes the instantaneous loss. Simulation results showed that the DES-based prediction strategy provided a similar hit ratio and cache update ratio to LFU. This might reflect the inability of a single expert to adapt to the entire popularity trend, so combining experts may improve performance.

Ben Hassine *et al.* [56] found that models based on multiple k best experts outperform single expert models because they can better handle different patterns and trends. Generally speaking, one expert cannot model all user behaviors accurately. Therefore, a forecaster model was designed and was responsible for selecting the best k expert that produced accurate prediction results. It can also change the experts after every period of time t to minimize the cumulative loss and adapt to changes in user behavior. Moreover, in [57], a forecaster using auto-regressive moving average (ARMA) experts was designed in order to minimize the MSE. ARMA combines the moving average and auto-regressive models. Famaey *et al.* [58] considered changing the LFU from frequency-based to future popularity-based and called it optimal-selection predictive least frequently used (OP-LFU). The input to OP-LFU is the previous accumulated number of perceived requests for content recorded every hour for a duration of 12 hours. The requested video popularity was then predicted using linear, power-law, exponential and Gaussian models, which are specialized for constant increase, steep changes, slow changes and S-shaped changes in popularity, respectively. The optimal model that described the evolution pattern was selected based on the MSE.

The main drawback of techniques in [8] and [55]–[58] is the process of selecting the best model, which requires the evaluation of all models in term of metrics such as the MSE. There is also the need to repeat the process individually for every specific time period and across all the contents, which results in long training periods and a slow caching procedure. For the purpose of reducing the training time, Hou *et al.* [59] proposed a transfer learning-based approach. Unlike traditional machine learning models which start learning from scratch when new training data are added, transfer learning tries to transfer the knowledge from prior relative tasks to a latter task accelerating the training process. Additionally, the authors designed a greedy algorithm to solve the caching optimization problem.

On the other hand, Li *et al.* [16] eliminated the need for training phase by leveraging the similarity of access patterns. They created the Pop Caching scheme, an online caching scheme that eliminated the need for a training phase. If the requested video is not available in the nearby node, the client is served by a more distant one. Then, based on its context vector, the cache decision is made. The context vector contains four historical requests values which mark a point in a space that is divided into hypercubes, where each hypercube has a value of future popularity. When the actual demand is revealed, the popularity value is updated, and the hypercube may split into smaller hypercubes. This scheme suffers from low performance in the starting phase because the prediction method has no previous knowledge about popularity patterns, but it gradually learns them.

2) MODELS BASED ON METADATA

The historical popularity of old videos can foreshadow their popularity in the near future. However, this is not

applicable to newly uploaded videos where there may be no records about previous popularity. To deal with recent videos, the video metadata and the user data such as watch time, shares, subscribers, and video age should be considered. For instance, Abdelkrim *et al.* [20] applied a hybrid multi-variance regression line that does not use historical view counts to predict popularity. A features vector was produced by preprocessing logs in the accounting and logging unit in the content delivery network. Furthermore, the authors tried to overcome the issues raised by using offline or online approaches. In offline, the computed coefficients are fixed and generated once from a training set, but in online, old parameters are discarded for every new time window. The proposed hybrid approach combines the two approaches by starting with the offline parameters then updating them using the exponential weighted moving average. Aloui *et al.* [60] used similar features to [20] and K-means algorithm to divide videos into five different popularity categories. Besides that, Tang *et al.* [4] implemented a caching scheme which forecasts the popularity of contents based on a continuously updated neural network with stateful features (dynamic features whose state must be tracked) and stateless features, which are almost stable.

As mentioned earlier, the number of viewers is usually used to denote videos popularity, but some researchers suggest other popularity measures such in [18] and [61]. Nogueira *et al.* [18] introduced expected priority and designed a caching scheme that favored TV shows with a greater expected priority, at the expense of others with lower expected priorities. Higher priorities map to higher expectations that an item is going to be requested in the future. The expected priorities were created using a random forest approach, which is suitable for online training, whereby an existing predictive model is improved using new data without fully retraining it. Moreover, Xing *et al.* [61] introduced accumulated network traffic (ANT) as a content popularity metric. Popular multimedia contents are responsible for the vast majority of network traffic, so the traffic for particular content can represent its popularity. The overall user preferences of each cell were modeled using neural networks because it is more beneficial to consider the interest of groups than individuals interests, which are stochastic and highly depended on external factors. Each neural network estimates the next ANT values via a label vector, which contains features for each video such as metadata with additional information about the corresponding data packets. During the prediction stage, different neural network models were designed to determine contents popularity in particular cell during multiple time slots, helping to eliminate the effect of variance produced by different temporal behaviors. During the caching operation, the content is delivered to the user. The ANT is then updated based on the newly generated traffic if it was cached. Otherwise, the ANT value is predicted and compared with contents within the cache. The major drawback of [61] is the long training time required for the neural network.

The aforementioned works focused on accessing the video popularity, but they did not consider improving the contents dissemination and caching placement. Hoiles *et al.* [62] exploited game-theory in improving contents dissemination. The framework considered two intertwined aspects. First, the video popularity was determined based on users tendencies. Then, game-theory was used for caching dissemination. Starting with popularity prediction, popular contents were analyzed using the theory of revealed preferences. The popularity prediction scheme takes video metadata into account (e.g. thumbnail, publisher, title, description, and user reaction) to estimate request probabilities which show the estimated demand for videos. The values were used as inputs for the game-theory caching algorithm to optimize edge caching based on user behavior. The game-theory approach is applied in game settings where players (servers in this case) are competing and attempting to drive the system towards equilibrium, where no additional action by players can enhance the objective function. This study revealed that real edge units can organize their caching algorithms in a dispersed manner as if a centralized management unit exists that they all follow. Further improvement in the prediction methodology can be achieved by considering feature selection and the co-correlation among subscribers, comments, and ratings.

Tanzil *et al.* [17] also considered finding the optimal placement of videos in the content distribution network. They exploited deep belief network to find users' playback patterns which were later used in determining content popularity. Then, a greedy-based online edge caching algorithm was formulated to allocate content with the objective of minimizing the content delivery cost. For the same purpose, Tanzil *et al.* [17] proposed an adaptive and online caching scheme that uses a mixed-integer linear program (MILP) to initialize the cache during low network load, based on the predicted content popularity and cellular network characteristics. The cache is then updated using S3LRU. Here, the prediction of popularity was achieved using an extreme learning machine (ELM). The built ELM is a single hidden-layer feed-forward neural network with random hidden layer parameters, and randomly initialized weights between the input and hidden layer. This requires less training time than many ML models because only the weights between the last hidden layer and the output layer are computed. The input vector contained the number of subscribers, last-day view count, and video thumbnail contrast and overexposure. It was selected after applying sequential wrapper feature selection on 54 features.

Popularity-based caching in cloud-based radio access networks (CRANs) was addressed in [63] and [64]. Chen *et al.* [63] formulated an optimization problem with a primary objective of maximizing the effective capacity to decide contents and their caching locations. The optimization equation involved contents popularity and periodic movement of users which were foreseen using a deep learning framework based on echo state networks (ESN). ESN solves the problem of long training time in recurrent neural

networks (RNN) by assigning random weights to edges between the input layer and hidden layer as well as between hidden layers. Then, only weights of the connections between the last hidden layer and the output are computed. The ESN algorithms were employed in baseband units (BBUs). For each user, two ESN models were used. The first one predicts probability of a user viewing a set of contents. The second ESN predicts the user's next location. Having predicted the anticipated locations and future requests, users with a similar distribution of requests were clustered and contents were ranked to decide what to store in the cloud. The contents ranking and clustering process for the substantial quantity of data were accelerated using the sublinear approach. Additionally, cache-enabled UAVs were added in [64] to CRANs and conceptor-based echo state networks (ESNs) was used to obtain the user-UAV pairs, the suitable UAVs' locations, and the contents to be cached.

However, in all the previously discussed works, the newly uploaded videos and the other videos are treated by using the same prediction algorithm. Meanwhile, new videos lack a lot of information especially statistical one making obtaining accurate prediction harder. It worth noting that new videos are relatively small compared to other videos but their popularity can grow dramatically in a short period of time. For this reason, Doan *et al.* [65] designed deep learning based model that deal with new video. The model uses C3D CNN architecture to extract high-level features from video raw data. The popularity of new videos is then determined based on the similarity between them and the older videos.

3) SOCIAL DYNAMICS

Social connections and user interactions can be expressed by social graphs, which consist of nodes and links. There are multiple types of social graphs, including friendship, interaction, latent, and following graphs [66]. In social dynamic models, social graphs are generated then utilized to predict the propagation of videos in the network and consequently their future popularity. Sengupta [67] proposed a framework for identifying patterns of views and shares using traces collected from Facebook. The collected traces were used to create an event tree for all videos, with users as the nodes of the tree and the connections as the events (i.e., views and shares). After creating the complex graph, it was simplified by removing nodes that were insignificant for video distribution. The final graph was used for classifying contents. However, the performance of the model was not evaluated based on cache-centric metrics.

Nwana *et al.* [68] merged the consensus and social approach. In the consensus approach, the request rate was predicted using the power law distribution model. Furthermore, content was kept in the cache if the view count was increasing (checked by applying a second derivative test) and there was a small time difference between the last access time and the time when the predicted popularity was high. In the social approach, a virus propagation model was applied to the latent social graph to estimate sharing probabilities and

future view counts. The implemented propagation model was a susceptible-infected model with two states: susceptible and infected. The susceptible state comprises users who have not view the video yet. After viewing the video, users are infected and share the video with others with some probability. The social graph does not include all the users, and the consensus approach compensates for the graph limitations.

He *et al.* [51] modeled the social propagation of content by using a susceptible-infected-recovery (SIR) model to improve prediction accuracy. SIR is often used to model the spreading of epidemics and comprises three states: susceptible, informed, and refractory. In the susceptible state, users are not notified of the content. Later, when users view the content, they move to the informed state, and are ready to decide whether or not to share it with their followers. If they share it, it appears on their news feed, and all their followers can view the video. Users in the informed state keep propagating the content to their friends. In contrast, if the content was viewed, but the viewer is not willing to share it, he/she moves to refractory state. Every user either shares the content or moves to the refractory state with different probabilities. Those probabilities are dynamically updated to reflect each user's interactions, and can be calculated by analyzing real traces and social connections. A discrete-time Markov chain which considers the SIR results is then designed to determine the view count.

Social graphs can produce accurate predictions by considering the social interactions of internet users. However, they suffer from two main shortcomings: graph complexity and dynamic changes. Complex social graphs increase computational costs and storage requirements. Therefore, graph sampling and crawling techniques have been proposed to reduce the complexity. Graph sampling techniques can create smaller representative graphs while maintaining the degree distribution. Moreover, the temporal and spatial dependence between different data items can be exploited for better compression. For example, Jin *et al.* [66] studied the spatial effect on the spreading of content and found that spreading is a function of distance. Wang *et al.* [69] investigated social groups formed by users sharing similar interests and discovered that unpopular videos tend to propagate in groups that are strongly connected socially. Dynamic changes, which also affect the reliability of social graphs, include changes in social links, behavior, and personal taste over time, as well as changes in the social graph due to additional users enrolling in the network.

B. CROSS-DOMAIN

Cross-domain models utilize features from external sources to increase the accuracy of prediction. One of the initial attempts to use cross-domain features is in [70]. Roy *et al.* [70] revealed that social streams such as Twitter learn about real-life events quickly, especially breaking news, before these events reach other websites or even TV channels. The authors found that the popularity originated on Twitter then spread to YouTube. Therefore, they used data from

Twitter to find videos that are likely to experience sudden popularity bursts on YouTube. The prediction process started by extracting popular topics from tweets, associating these topics with YouTube videos, and comparing the popularity computed using Twitter data with their popularity on YouTube. A large difference would indicate a sudden burst of popularity expected for the YouTube video.

A similar study was conducted on so-like YouTube and Twitter sites famous among Chinese users: Youku and Tencent Weibo. Wang *et al.* [71] built a proactive caching framework for Youku videos, using the features of microblogs. They found a correlation between the popularity distribution in Weibo and video views in Youku. Moreover, the spreading speed of video links in microblogs was faster than the actual video views on a video sharing site. This result in a time lag between both events that could allow the more timely and proactive deployment of videos. They implemented two three-layer feed-forward neural networks: one to predict the number of potential viewers and the other to show the geographic distribution of viewers. The features of the first neural network were assigned weights by Pearson correlation. For the second neural network, the geographic distribution of Weibo users who have published microblogs with video links was used to estimate the spatial data of all viewers in Youku. The output was a vector illustrating geographic popularity in different regions in China and foreign countries. Finally, the obtained results were used to improve the caching and replication strategy in these regions.

However, not all the information in social media is useful. Lobzhanidze and Zeng [72] addressed the noise issue in social networks resulted from worthless conversations leaving only 3.6% for news sharing. Compared to social media, the mainstream media contains much less amount of information which can be used directly in recognizing trends faster and more accurate. The regional popularity can be determined by making the ISP utilizes the country mainstreams media such as CNN and BBC for English-speaking countries. The process commenced with the creation of a list of popular topics using latent Dirichlet allocation (LDA) to obtain the most frequent well-defined words. Frequent pattern mining was then used to find other words, including the names of famous people, and this was applied to the document titles with the most popular topics. Every popular word was then used as a query in YouTube search and only the top-five ranking videos were studied in detail. This is because the remaining results accounted for less than 5% of the click rate. The selected caching objects were evaluated by their subscribers and reputation system. The reputation system was generated by monitoring the views of all the selected contents. For content eviction, the comparison metrics were based on the current popularity of each video, the popularity of the topic, and the duration of popularity for the video category.

However, the aforementioned works used features from external sources only and did not attempt to model content propagation in videos streaming sites. Generally speaking,

the ecosystem of videos streaming site cannot help in spreading the video widely as in social media. Thus, dynamic social graph created from social networks can be utilized to approximate the popularity in video sharing site. Soysa *et al.* [73] proposed creating propagation model for Facebook users and use it on YouTube videos that lack the user interaction information. The intuition is content viewed by influential users is highly expected to propagate widely. Facebook data about users and their friends were obtained using Scraper. Then, the fast threshold spread model (FTSM) was used to create the social graph. FTSM is a propagation model that provides a quick approximation of the independent cascade model (ICM). Data reflecting user influence and activity levels were used as inputs to the model. Similarly, in [74], video propagation pattern in Youtube was predicted using Facebook users' data. The advanced independent cascade model (AICM), which extend ICM to include the effect of users' influential power, is proposed to establish the social graph and anticipate the spread of videos. However, the graph modeling of large networks results in sophisticated graphs and long execution times, which is not ideal for caching schemes. Therefore, Soysa *et al.* [73] recommended using a small network that describes the larger network, but generating the optimal size for the graph remains challenging.

C. COMPARISON AMONG POPULARITY-BASED CACHING ALGORITHMS

An efficient video caching scheme must be popularity-based, meaning that it should access the future video popularity and leverage it in making appropriate caching decisions. There are many popularity-based caching algorithms adopted in the field of video caching. We divide them, according to the features used, into two main classes: models based on 1) single domain data and 2) cross-domain data. Single domain models can be divided further into three subclasses: models based on 1) popularity evolution trends, 2) social dynamics, and 3) video metadata.

- 1) Popularity evolution trends: A popularity evolution trend illustrates how the video is evolving over time, and it is expressed by historical popularity values. In this category, models estimate the future popularity by capturing temporal correlations in the sequence of popularity values. It also tries to understand videos evolution patterns. There are different video popularity evolution patterns. Figueiredo *et al.* [75] distinguished four evolution trends by applying a time series clustering algorithm. Richier *et al.* [76] developed seven mathematical models and found that more than 92% of videos can be associated with one of the models, with a mean error rate less than 5%. The ability of the model to capture changes in the video evolving trend can highly impact the prediction performance which posts a challenge for this category of models [77]. Thus, recent studies attempted to learn the video popularity trend, then used it in improving the prediction accuracy. For example, in [78], two-step learning approach was

proposed. In the first step, the video is classified into one of four trends. Then, a trend-specific random forest model is applied.

- 2) Models based on metadata: The previous category depends on the video popularity values and does not explain why a video reaches a certain popularity value. Video metadata provides a lot of useful information that can show why the video become popularity such as: watch time, shares, subscribers, view count, and channel information. Thus, models based on metadata can improve the prediction accuracy especially for the young videos, where information from popularity evolution curve is limited or not even applicable. Moreover, video metadata is stored by video providers and can be easily retrieved by the public. It also requires low processing time, because it is recorded and tracked by the video sharing site. This results in low complexity of the preprocessing stage.
- 3) Social dynamics: Social behaviors have remarkably reformed how videos are spreading, where users nowadays are not only watching videos but also globally propagate them through social networks. Social dynamics models extremely depend on predicting the spreading pattern using information about social relationships, users connections, and users' influence power. Then, based on how the video is propagating, its future popularity can be derived. It has been found in [67] and [69] that historical popularity values and metadata may not always represent the true future popularity, whereas social dynamics among users can produce much more precise estimations. But, despite the high accuracy of social models, they suffer from two main issues: graph complexity and dynamic changes. Massive social networks require complex social graphs which correspond to high computational costs and storage capacity. Moreover, dynamic changes in the social network can dramatically impact the prediction reliability, such changes include variation in social connection, users' relationship, and personal taste. Another major problem in this category is obtaining the social data. In fact, information regarding social connections is not provided to the public on video streaming sites such as YouTube.
- 4) Cross-domain models use features from external sources to improve the prediction performance. They extract information related to the video from multiple domains (e.g., social media) and combine their knowledge to foreseen video popularity in another domain (e.g., video sharing site). The potential benefits of cross-domain models can be significantly noticeable when the video is propagating through web services, where they can yield better prediction accuracy than the single domain methods [14]. But, cross-domain methods require processing features from the video site and other sources which can result in a high computational cost. Moreover, not all the information in social media

is useful, where the majority are representing worthless conversations.

VI. OPEN RESEARCH AREAS

ML-based prediction is a mature field in data science, but its integration with caching networks involves plenty of intricacies that needs to be handled. This section highlights some of the open research areas for proactive caching including caching based on big data tools, green caching, resource management, and caching enhancements.

A. CACHING BASED ON BIG DATA TOOLS

The accurate tracking of spatio-temporal user behavior is necessary to understand user mobility, habits, interests, and social influence. This requires the analysis and processing of extraordinary quantities of data to discover hidden patterns in user activities and similarities among groups in a region. Sophisticated models are needed to take into account the high dimensionality of input vectors, the large number of users, and the even larger amount of contents. Research has therefore focused on leveraging big data analytics in cache-enabled networks [2], [79]. Baştuğ *et al.* [79] exploited the big data platform integrated with the network operator to collect and process user data, and to execute content prediction algorithms in parallel for faster cache dissemination. Furthermore, Hadoop's distributed data processing engine was used in [2], to analyze user behavior based on large amounts of streaming data.

Big data analytical tools include stochastic modeling, data mining, ML, and data reduction. Stochastic modeling captures the explicit features and dynamics of the data traffic. Data mining finds patterns in data. ML links inputs with outputs [80]. Data reduction methods handle the complexity associated with big data as denoted by 4V: volume, velocity, veracity, and variety [81]. Hadoop and Spark are frequently used open source big data platforms. They distribute the computational load among machines to execute them simultaneously instead of the traditional serial processing. Hadoop uses MapReduce for tasks parallelization. MapReduce consists of two phases: map and reduce. First, the workload is split into smaller tasks and distributed among the mappers. Each mapper is responsible for processing chunks of the data. When the map phase is finished, the reduce phase begins and combines the results to produce the desired output. MapReduce cannot run on memory and data must be written to disk. In contrast, Spark can perform in-memory processing, resulting in remarkably fast processing that can handle real-time applications. It is also a unified API where streaming, structured query language (SQL), machine learning and graphing tools are combined [82].

B. GREEN CACHING NETWORK

Green caching attracts a significant amount of research attention especially with the 5G aspiration to increase the network capacity by 1000 times beyond 4G, while maintaining or reducing power consumption [13]. The goal of green

caching is to minimize overall network energy while serving user demands, and satisfying network resource capacities. Smart caching mechanisms based on popularity prediction can reduce power consumption by transferring on-demand contents to edges. However, smart algorithms are likely to be very complex, resulting in high computational and energy costs. It is important to note that reducing power consumption not only requires the prediction of popular contents but also the intelligent selection of where to place them. In cellular networks, the smaller the cell, the lower the coverage area and power consumption. Liu and Yang [83] found that that content placement in pico-base stations improves energy efficiency in local caching compared to caching in macro-base stations. However, pico-base stations have limited backhaul capacity. Moreover, optimal transmission path selection can reduce transmission power. For example, in information-centric-network (ICN), the content should pass through the minimum number of hops to reduce energy costs. Dehghan *et al.* [84] proposed a cache-aware routing scheme to compute the paths with the minimum transmission cost based on content demand and the caching capabilities of the network.

Another concern for green caching is the trade-off between content caching costs and transmission costs. To bring the content close to users, multiple copies must be inserted in various caches. However, this might increase the costs of content storage for the sake of reducing the download time, whereas placing the content further away would reduce the need for duplication [40].

C. MANAGEMENT OF CACHING RESOURCES

Proactive caching helps to manage the available storage by mitigating the number of contents worth caching. However, the continuous monitoring of cached contents is required because correct prediction cannot be guaranteed throughout the life of the regime. The prediction accuracy may initially be high, but errors may appear later. To overcome the impact of errors and to ensure the robustness and stability of the prediction model, online learning schemes need to be applied [15], [85]. Furthermore, CPU and RAM capabilities are limited caching resources. Even if the prediction takes milliseconds to predict the popularity of a single video, minutes will be needed to process the large amount of data and requests [52]. Therefore, big data frameworks, data selection, and sublinear filtration are needed before applying the algorithm. Sublinear algorithms read parts of the input and provide an approximate solution during the affordable processing time. The computation complexity of such algorithms is sublinear in time or space with respect to the input size, which is the main reason for the interest in this emerging field of data science [86].

D. PERFORMANCE ENHANCEMENT AND TRADE-OFF

Maximizing or reaching optimal caching performance in arbitrary networks remains an NP-hard problem [17]. Several methods and technologies have been integrated with

proactive caching to improve performance including the following:

1) MULTICASTING

Instead of serving users individually, multicasting broadcasts the content to multiple users simultaneously, resulting in improving the QoE and reducing the required bandwidth. It increases the global network gain for live streaming videos, but typical multicasting is not feasible for video-on-demand (VoD) because streaming periods occur asynchronously at unpredictable times [21]. In order to leverage single transmission to multiple end users for VoD applications, a coded caching scheme known as coded multicast can be used. In coded multicast, the transmission of various files is achieved by XORing different files and sending them in one packet [87]. Nevertheless, coding complexity increases dramatically with the number of users, resulting in computational resource consumption and delayed delivery. Asghari *et al.* [87] attempted to decrease the communication load by reducing the encoding bits while catering for all clients' requests.

2) DEVICE TO DEVICE (D2D) COMMUNICATION

D2D can manage local broadcasting effectively because it can leverage the proliferation of users' equipment by exploiting the storage of the devices for caching purposes. Therefore, if parts of a popular content are stored in nearby devices, requests can be achieved through D2D communications instead of base stations [88], [89]. In [90], caching-based D2D communication was analyzed in real-life scenarios. It was found that D2D in caching networks can improve throughput by 10^1 to 10^2 times at outage probabilities of 0.01 to 0.1. A network involving D2D communications used in different applications is shown in Fig. 7.

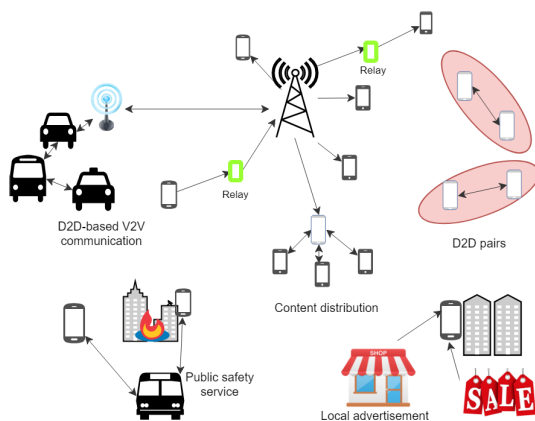


FIGURE 7. An example of network supporting D2D communication.

Selecting which content to cache in SBSs and user equipment is not trivial, but it can be achieved using popularity prediction. Moreover, user mobility may corrupt the transmission when user equipment moves out of the coverage area, especially for large content items. Therefore, it is better

to store large content files in SBSs. A better approach is to anticipate user movement and use it to improve performance [88]. D2D communication consumes user equipment battery power and storage space, so users tend to delete files that are no longer needed. The caching scheme should deal with this possibility to maintain the caching performance.

3) FEMTOCACHING

Small cells will probably have an important role in the support of video streaming [91]. Recently, many operators have considered small cell deployment, including microcells, picocells and femtocells in indoor/outdoor regions, residential districts, and hotspot areas. This is because small cells are an effective solution for offloading traffic, expanding coverage, and increasing network capacity. Fig. 8 shows a comparison among the small cell types in term of coverage area and power consumption.

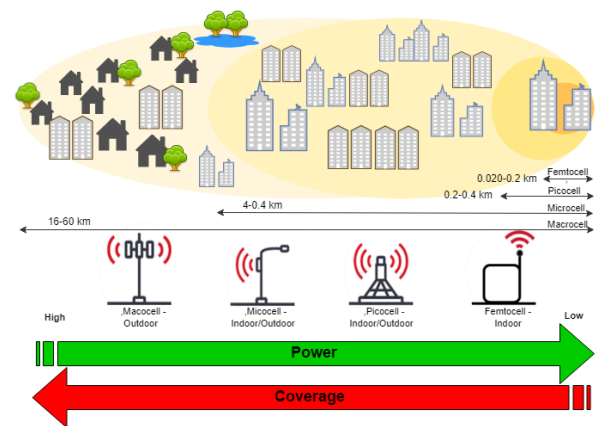


FIGURE 8. Small cell types.

Femtocells have minimal power consumption and limited coverage [92], [93]. They are deployed in the heterogeneous network to handle local communication and to reduce the network load. They can easily be placed in houses and workplaces to boost network speed and user QoE. However, they suffer from limited coverage capabilities and expensive back-haul connection [94]. Proactively and intelligently caching selected content in femtocells will help to overcome these limitations [93].

VII. CONCLUSION

The unprecedented growth in data has directed the next generation of mobile networks (5G) towards cognitive and proactive behaviors. It is more beneficial to make caching decisions based on the future popularity of contents rather than reacting to popularity variations when they occur. Videos are unlike other contents because they require high data transfer rates and large storage which make popularity prediction highly essential for videos caching. In this survey, we have summarized recent work in the field of video caching, and have described video features and the evaluation metrics of caching schemes. We have analyzed the challenges of

cache-enabled networks and discussed future research directions. In conclusion, although ML and prediction algorithms allow the caching network to uncover relationships among instances, predict dynamic behaviors, and make decisions intelligently without human intervention, proactive caching is still in its initial states. More research is needed to find optimal ways to incorporate proactive caching with big data methods and network technologies. Moreover, optimal solutions that maximize speed and minimize cost and power are challenging due to the remaining trade-offs among these three key parameters.

REFERENCES

- [1] (Sep. 2016). *The Need for Mobile Speed: How Mobile Latency Impacts Publisher Revenue*. [Online]. Available: <https://www.doubleclickbygoogle.com/articles/mobile-speed-matters/>
- [2] E. Zeydan *et al.*, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.
- [3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
- [4] L. Tang, Q. Huang, A. Puntambekar, Y. Vigfusson, W. Lloyd, and K. Li, "Popularity prediction of Facebook videos for higher quality streaming," in *Proc. USENIX Annu. Tech. Conf. USENIX*, 2017, pp. 1–15.
- [5] K. Mokhtarian and H.-A. Jacobsen, "Caching in video CDNs: Building strong lines of defense," in *Proc. ACM 9th Eur. Conf. Comput. Syst.*, 2014, p. 13.
- [6] A. Kishore, "Video-optimized caching: New challenges for delivery networks," Heavy reading, New York, NY, USA, White Paper, Mar. 2012, pp. 1–7.
- [7] *Mobile Video Delivery With Hybrid ICN*, Cisco, San Jose, CA, USA, 2017, pp. 1–14.
- [8] N. Ben Hassine, D. Marinca, P. Minet, and D. Barth, "Caching strategies based on popularity prediction in content delivery networks," in *Proc. IEEE 12th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2016, pp. 1–8.
- [9] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1002–1026, 2nd Quart., 2016.
- [10] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: A survey," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, p. 33, 2017.
- [11] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 954–1001, 2nd Quart., 2017.
- [12] R. Mahapatra, Y. Nijssure, G. Kaddoum, N. Ul Hassan, and C. Yuen, "Energy efficiency tradeoff mechanism towards wireless green communication: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 686–705, 1st Quart., 2016.
- [13] S. Buzzi, C.-L. I. T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, "A survey of energy-efficient techniques for 5G networks and challenges ahead," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 697–709, Apr. 2016.
- [14] A. Tatar, M. D. de Amorim, S. Fdida, and P. Antoniadis, "A survey on predicting the popularity of Web content," *J. Internet Services Appl.*, vol. 5, no. 1, p. 8, 2014.
- [15] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1790–1821, 3rd Quart., 2017.
- [16] S. Li, J. Xu, M. van der Schaar, and W. Li, "Popularity-driven content caching," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2016, pp. 1–9.
- [17] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach," *IEEE Access*, vol. 5, pp. 5870–5881, 2017.
- [18] J. Nogueira, D. Gonzalez, L. Guardalben, and S. Sargento, "Over-the-top catch-up TV content-aware caching," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 1012–1017.
- [19] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, "Video popularity dynamics and its implication for replication," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1273–1285, Aug. 2015.
- [20] E. Ben Abdelkrim, M. A. Salahuddin, H. Elbiaze, and R. Glitho, "A hybrid regression model for video popularity-based cache replacement in content delivery networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.
- [21] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.
- [22] Z. Wang *et al.*, "Propagation-based social-aware multimedia content distribution," *ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)*, vol. 9, no. 1s, p. 52, 2013.
- [23] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting video server selection strategies in the YouTube CDN," in *Proc. IEEE 31st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2011, pp. 248–257.
- [24] C. Koch, G. Krupii, and D. Hausheer, "Proactive caching of music videos based on audio features, mood, and genre," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 100–111.
- [25] W. Hoiles, A. Aprem, and V. Krishnamurthy, "Engagement and popularity dynamics of YouTube videos and sensitivity to meta-data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 7, pp. 1426–1437, Jul. 2017.
- [26] T. Trzciński and P. Rokita, "Predicting popularity of online videos using support vector regression," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2561–2570, Nov. 2017.
- [27] S. Ouyang, C. Li, and X. Li, "A peek into the future: Predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 3026–3033, 2016.
- [28] F. Figueiredo, F. Benevenuto, and J. M. Almeida, "The tube over time: Characterizing popularity growth of YouTube videos," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 745–754.
- [29] B. Su, Y. Wang, and Y. Liu, "A new popularity prediction model based on lifetime forecast of online videos," in *Proc. IEEE Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, Sep. 2016, pp. 376–380.
- [30] M. Garetto, E. Leonardi, and S. Traverso, "Efficient analysis of caching strategies under dynamic content popularity," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 2263–2271.
- [31] S. Yu and S. Kak. (2012). "A survey of prediction using social media." [Online]. Available: <https://arxiv.org/abs/1203.1647>
- [32] J. Miao and L. Niu, "A survey on feature selection," *Procedia Comput. Sci.*, vol. 91, pp. 919–926, Jan. 2016.
- [33] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.
- [34] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process.*, vol. 5, no. 2, pp. 1–11, 2015.
- [35] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *J. Mach. Learn. Res.*, vol. 10, pp. 2935–2962, Dec. 2009.
- [36] A. C. Cameron and F. A. Windmeijer, "An R-squared measure of goodness of fit for some common nonlinear regression models," *J. Econ.*, vol. 77, no. 2, pp. 329–342, 1997.
- [37] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.
- [38] M. Neishaboori *et al.*, "Implementation and evaluation of mobile-edge computing cooperative caching," M.S. thesis, School Sci., Aalto Univ., Helsinki, Finland, 2015.
- [39] H. Shi, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, "Fairness in wireless networks: Issues, measures and challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 5–24, 1st Quart., 2014.
- [40] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 122–129, Nov. 2014.
- [41] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.
- [42] Z. Tong, Y. Xu, T. Yang, and B. Hu, "Quality-driven proactive caching of scalable videos over small cell networks," in *Proc. IEEE 12th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2016, pp. 90–96.

- [43] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.
- [44] J. Greengrass, J. Evans, and A. C. Begen, "Not all packets are equal, part I: Streaming video coding and SLA requirements," *IEEE Internet Comput.*, vol. 13, no. 1, pp. 70–75, Jan. 2009.
- [45] B. Shulman, A. Sharma, and D. Cosley, "Predictability of popularity: Gaps between prediction and understanding," in *Proc. ICWSM*, 2016, pp. 348–357.
- [46] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, You Tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 1–14.
- [47] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Commun. ACM*, vol. 53, no. 8, pp. 80–88, 2010.
- [48] Z. Xiaoqiang, Z. Min, and W. Muqing, "An in-network caching scheme based on betweenness and content popularity prediction in content-centric networking," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.
- [49] A. Ioannou and S. Weber, "Exploring content popularity in information-centric networks," *China Commun.*, vol. 12, no. 7, pp. 13–22, 2015.
- [50] Y. Zhang, X. Tan, and W. Li, "PPC: Popularity prediction caching in ICN," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 5–8, Jan. 2017.
- [51] S. He, H. Tian, and X. Lyu, "Edge popularity prediction based on social-driven propagation dynamics," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1027–1030, May 2017.
- [52] H. Nakayama, S. Ata, and I. Oka, "Caching algorithm for content-oriented networks using prediction of popularity of contents," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1171–1176.
- [53] J. Li, S. Hong, S. Xia, and S. Luo, "Neural network based popularity prediction for IPTV system," *J. Netw.*, vol. 7, no. 12, pp. 2051–2056, 2012.
- [54] G. Silvestre, S. Monnet, D. Buffoni, and P. Sens, "Predicting popularity and adapting replication of Internet videos for high-quality delivery," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2013, pp. 412–419.
- [55] N. Ben Hassine, D. Marinca, P. Minet, and D. Barth, "Popularity prediction in content delivery networks," in *Proc. IEEE 26th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Aug./Sep. 2015, pp. 2083–2088.
- [56] N. Ben Hassine, D. Marinca, P. Minet, and D. Barth, "Expert-based on-line learning and prediction in content delivery networks," in *Proc. IEEE Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Mar. 2016, pp. 182–187.
- [57] N. Ben Hassine, R. Milocco, and P. Minet, "ARMA based popularity prediction for caching in content delivery networks," in *Proc. IEEE Wireless Days*, Mar. 2017, pp. 113–120.
- [58] J. Famaey, F. Iterbeke, T. Wauters, and F. De Turck, "Towards a predictive cache replacement strategy for multimedia content," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 219–227, Jan. 2013.
- [59] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," *Int. J. Commun. Syst.*, vol. 31, no. 11, p. e3706, 2018.
- [60] M. Aloui, H. Elbiaze, R. Glitho, and S. Yangui, "Analytics as a service architecture for cloud-based CDN: Case of video popularity prediction," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–4.
- [61] L. Xing, Z. Zhang, H. Lin, and F. Gao, "Content centric network with label aided user modeling and cellular partition," *IEEE Access*, vol. 5, pp. 12576–12583, 2017.
- [62] W. Hoiles, O. N. Gharehshiran, V. Krishnamurthy, N. D. Dao, and H. Zhang, "Adaptive caching in the YouTube content distribution network: A revealed preference game-theoretic learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 1, no. 1, pp. 71–85, Mar. 2015.
- [63] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.
- [64] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.
- [65] K. N. Doan, T. Van Nguyen, T. Q. S. Quek, and H. Shin, "Content-aware proactive caching for backhaul offloading in cellular network," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3128–3140, May 2018.
- [66] L. Jin, Y. Chen, T. Wang, P. Hui, and A. V. Vasilakos, "Understanding user behavior in online social networks: A survey," *IEEE Commun. Mag.*, vol. 51, no. 9, pp. 144–150, Sep. 2013.
- [67] S. Sengupta, "Predicting social dynamics based on network traffic analysis for CCN/ICN management," in *Proc. 9th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2017, pp. 584–585.
- [68] A. O. Nwana, S. Avestimehr, and T. Chen, "A latent social approach to YouTube popularity prediction," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 3138–3144.
- [69] Z. Wang, J. Liu, and W. Zhu, "Social-aware video delivery: Challenges, approaches, and directions," *IEEE Netw.*, vol. 30, no. 5, pp. 35–39, Sep./Oct. 2016.
- [70] S. D. Roy, T. Mei, W. Zeng, and S. Li, "Towards cross-domain learning for social video popularity prediction," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1255–1267, Oct. 2013.
- [71] Z. Wang, L. Sun, C. Wu, and S. Yang, "Enhancing Internet-scale video service deployment using microblog-based prediction," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 775–785, Mar. 2015.
- [72] A. Lobzhanidze and W. Zeng, "Proactive caching of online video by mining mainstream media," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.
- [73] D. A. Soysa, D. G. Chen, O. C. Au, and A. Bermak, "Predicting YouTube content popularity via Facebook data: A network spread model for optimizing multimedia delivery," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2013, pp. 214–221.
- [74] D. A. Soysa, O. C. Au, L. Sun, L. Xu, J. Li, and D. G. Chen, "Advanced independent cascade model for YouTube content propagation in Facebook," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2013, pp. 481–485.
- [75] F. Figueiredo, J. M. Almeida, M. A. Gonçalves, and F. Benevenuto, "Trendlearner: Early prediction of popularity trends of user generated content," *Inf. Sci.*, vols. 349–350, pp. 172–187, Jul. 2016.
- [76] C. Richier, R. Elazouzi, T. Jimenez, E. Altman, and G. Linares, "Predicting popularity dynamics of online contents using data filtering methods," in *Proc. IEEE 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 31–38.
- [77] C. Li, J. Liu, and S. Ouyang, "Characterizing and predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 1630–1641, 2016.
- [78] C. Zhu, G. Cheng, and K. Wang, "Big data analytics for program popularity prediction in broadcast TV industries," *IEEE Access*, vol. 5, pp. 24593–24601, 2017.
- [79] E. Baştuğ et al., "Big data meets telcos: A proactive caching perspective," *J. Commun. Netw.*, vol. 17, no. 6, pp. 549–557, Dec. 2015.
- [80] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Res.*, vol. 2, no. 3, pp. 87–93, 2015.
- [81] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Commun. Mag.*, vol. 53, no. 10, pp. 190–199, Oct. 2015.
- [82] M. Zaharia et al., "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.
- [83] D. Liu and C. Yang, "Energy efficiency of downlink networks with caching at base stations," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 907–922, Apr. 2016.
- [84] M. Dehghan et al., "On the complexity of optimal request routing and content caching in heterogeneous cache networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1635–1648, Jun. 2017.
- [85] J. Walter, A. Di Marco, S. Spinner, P. Inverardi, and S. Kounev, "Online learning of run-time models for performance and resource management in data centers," in *Self-Aware Computing Systems*. Cham, Switzerland: Springer, 2017, pp. 507–528.
- [86] A. Czumaj and C. Sohler, "Sublinear-time algorithms," in *Property Testing*. Berlin, Germany: Springer, 2010, pp. 41–64.
- [87] S. M. Asghari, Y. Ouyang, A. Nayyar, and A. S. Avestimehr, (2017). "An approximation algorithm for optimal coded multicast in cache networks." [Online]. Available: <https://arxiv.org/abs/1710.10718>
- [88] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001–5015, Aug. 2017.
- [89] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
- [90] M.-C. Lee, M. Ji, A. F. Molisch, and N. Sastry, (2018). "Performance of caching-based D2D video distribution with measured popularity distributions." [Online]. Available: <https://arxiv.org/abs/1806.05380>

- [91] V. A. Siris and D. Dimopoulos, "Multi-source mobile video streaming with proactive caching and D2D communication," in *Proc. IEEE 16th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2015, pp. 1–6.
- [92] *Small Cell Network White Paper*, Ericsson, Huawei, Shenzhen, China, Nov. 2016.
- [93] E. Baştuğ, M. Bennis, and M. Debbah, "Proactive caching in 5G small cell networks," in *Towards 5G: Applications, Requirements and Candidate Technologies*. New York, NY, USA: Wiley, 2016, pp. 78–98.
- [94] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

HUDA S. GOIAN received the B.Sc. degree in electrical engineering and the M.Sc. degree in electrical and computer engineering from Khalifa University, United Arab Emirates, in 2016 and 2018, respectively. Her current research interests include data analysis, artificial intelligence, wireless communications, and machine learning.



machine learning, intrusion detection, big data analytics, autonomous and connected vehicles, and knowledge discovery in various applications.

OMAR Y. AL-JARRAH received the B.S. degree in computer engineering from Yarmouk University, Jordan, in 2005, the M.S. degree in computer engineering from The University of Sydney, Sydney, Australia, in 2008, and the Ph.D. degree in electrical and computer engineering from Khalifa University, United Arab Emirates, in 2016. He is currently a Postdoctoral Fellow with the Warwick Manufacturing Group, The University of Warwick, U.K. His main research interests include



sor with Khalifa University and a Visiting Reader (Associate Professor) with the Faculty of Engineering, University of Surrey, U.K. His research interests include wireless communications, optical communications, the Internet of Things with emphasis on battery-less devices, and machine learning. He is also a member of the Mohammed Bin Rashid Academy of Scientists. He is currently an Area Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. He served as a Senior Editor for IEEE COMMUNICATIONS LETTERS, as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, and as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

SAMI MUHAIDAT (M'08–SM'11) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, in 2006. From 2007 to 2008, he was an NSERC Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Canada. From 2008 to 2012, he was an Assistant Professor with the School of Engineering Science, Simon Fraser University, BC, Canada. He is currently an Associate Profes-



YOUSOF AL-HAMMADI received the bachelor's degree in computer engineering from Khalifa University of Science and Technology (previously known as Etisalat College of Engineering), Abu Dhabi, United Arab Emirates, in 2000, the M.Sc. degree in telecommunications engineering from the University of Melbourne, Australia, in 2003, and the Ph.D. degree in computer science and information technology from the University of Nottingham, Nottingham, U.K., in 2009. He is currently Acting Dean of Graduate Studies and Assistant Professor in the Electrical and Computer Engineering Department, Khalifa University of Science and Technology. His main research interests include the area of information security, which includes intrusion detection, botnet/bots detection, viruses/worms detection, artificial immune systems, artificial intelligence, machine learning, RFID security, and mobile security.



PAUL YOO (SM'13) held academic/research positions with Defence Academy of the U.K., Cranfield; USyd, Sydney; and the Korea Advanced Institute of Science and Technology (KAIST), South Korea. He is currently with CSIS, Birkbeck College, University of London. He is a Visiting Professor with The University of Sydney and KAIST. He has amassed more than 60 prestigious journal and conference publications. He was a recipient of more than U.S. \$2.3 million in project funding. He received a number of prestigious international and national awards for his research in advanced data analytics, machine learning, and secure systems research, notably the IEEE Outstanding Leadership Award, the Capital Markets CRC Award, the Emirates Foundation Research Award, and the ICT Fund Award. He was a recipient of the prestigious Samsung Award for research to protect the Internet of Things devices. He also serves as an Editor for the IEEE COMML and the *Journal of Big Data Research* (Elsevier).



MEHRDAD DIANATI was a Professor with the 5G Innovation Centre, University of Surrey. He was a Senior Software/Hardware Developer and the Director of research and development with the industry for more than nine years. He has been involved in a number of national and international projects as the Project Leader and the Work-Package Leader for recent years. He is currently a Professor of autonomous and connected vehicles with the Warwick Manufacturing Group, The University of Warwick, and a Visiting Professor with the 5G Innovation Centre, University of Surrey. He frequently provides voluntary services to the research community in various editorial roles, for example, he has served as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *IET Communications*, and *Journal of Wireless Communications and Mobile* (Wiley).

...